CLAIMS:

1. A method for utilizing bits in an illegal op code in order to not increase the number of bits required to represent each instruction comprising the steps of:

fetching a plurality of instructions from a memory;

re-encoding one or more illegal op codes of one or more instructions into a single illegal op code if said one or more instructions comprise illegal op codes that are a member of a group of illegal op codes;

pre-decoding a fetched instruction that does not have an illegal op code; and

re-encoding one or more fields of said pre-decoded instruction into a previously illegal op code which was re-mapped into said single illegal op code.

2. The method as recited in claim 1, wherein said pre-decoding produces additional pre-decoded information, wherein said re-encoded pre-decoded instruction is configured to store said additional pre-decoded information in said previously illegal op code.

3. The method as recited in claim 2, wherein said additional pre-decoded information comprises a carry-out field.

4. The method as recited in claim 3, wherein said carry-out field is associated with a fetched branch instruction.

5. A processor, comprising:

an instruction cache configured to fetch a plurality of instructions; and

a logic unit coupled to said instruction cache configured to re-encode one or more illegal op codes of one or more instructions into a single illegal op code if said one or more instructions comprise illegal op codes that are a member of a group of illegal op codes, wherein said logic unit is further configured to pre-decode a fetched instruction that does not have an illegal op code, wherein said logic unit is further configured to re-encode one or more fields of said pre-decoded instruction into a previously illegal op code which was re-mapped into said single illegal op code.

6. The processor as recited in claim 5, wherein said pre-decoding produces additional pre-decoded information, wherein said re-encoded pre-decoded instruction is configured to store said additional pre-decoded information in said previously illegal op code.

7. The processor as recited in claim 6, wherein said additional pre-decoded information comprises a carry-out field.

8. The processor as recited in claim 7, wherein said carry-out field is associated with a fetched branch instruction.

9. A processor, comprising:

means for fetching a plurality of instructions from a memory;

means for re-encoding one or more illegal op codes of one or more instructions into a single illegal op code if said one or more instructions comprise illegal op codes that are a member of a group of illegal op codes;

means for pre-decoding a fetched instruction that does not have an illegal op code; and

means for re-encoding one or more fields of said pre-decoded instruction into a previously illegal op code which was re-mapped into said single illegal op code.

10. The processor as recited in claim 9, wherein said pre-decoding produces additional pre-decoded information, wherein said re-encoded pre-decoded instruction is configured to store said additional pre-decoded information in said previously illegal op code.

11. The processor as recited in claim 10, wherein said additional pre-decoded information comprises a carry-out field.

12. The processor as recited in claim 11, wherein said carry-out field is associated with a fetched branch instruction.

13. A system, comprising:

a memory configured to store instructions;

an instruction cache coupled to said memory, wherein said instruction cache is configured to fetch a plurality of instructions from said memory; and

a logic unit coupled to said instruction cache configured to re-encode one or more illegal op codes of one or more instructions into a single illegal op code if said one or more instructions comprise illegal op codes that are a member of a group of illegal op codes, wherein said logic unit is further configured to pre-decode a fetched instruction that does not have an illegal op code, wherein said logic unit is further configured to re-encode one or more fields of said pre-decoded instruction into a previously illegal op code which was re-mapped into said single illegal op code.

14. The system as recited in claim 13, wherein said pre-decoding produces additional pre-decoded information, wherein said re-encoded pre-decoded instruction is configured to store said additional pre-decoded information in said previously illegal op code.

15. The system as recited in claim 14, wherein said additional pre-decoded information comprises a carry-out field.

16. The system as recited in claim 15, wherein said carry-out field is associated with a fetched branch instruction.